

=====

Title: MS GW-BASIC INTERPRETER UNDER
MS-DOS USER GUIDE

Newsletter Code: 4034491 D

Date: 25th June 1987

Publication Code: 4034490 C (Ø)

Previous Newsletters: None

This Newsletter provides updated pages for the subject publication.

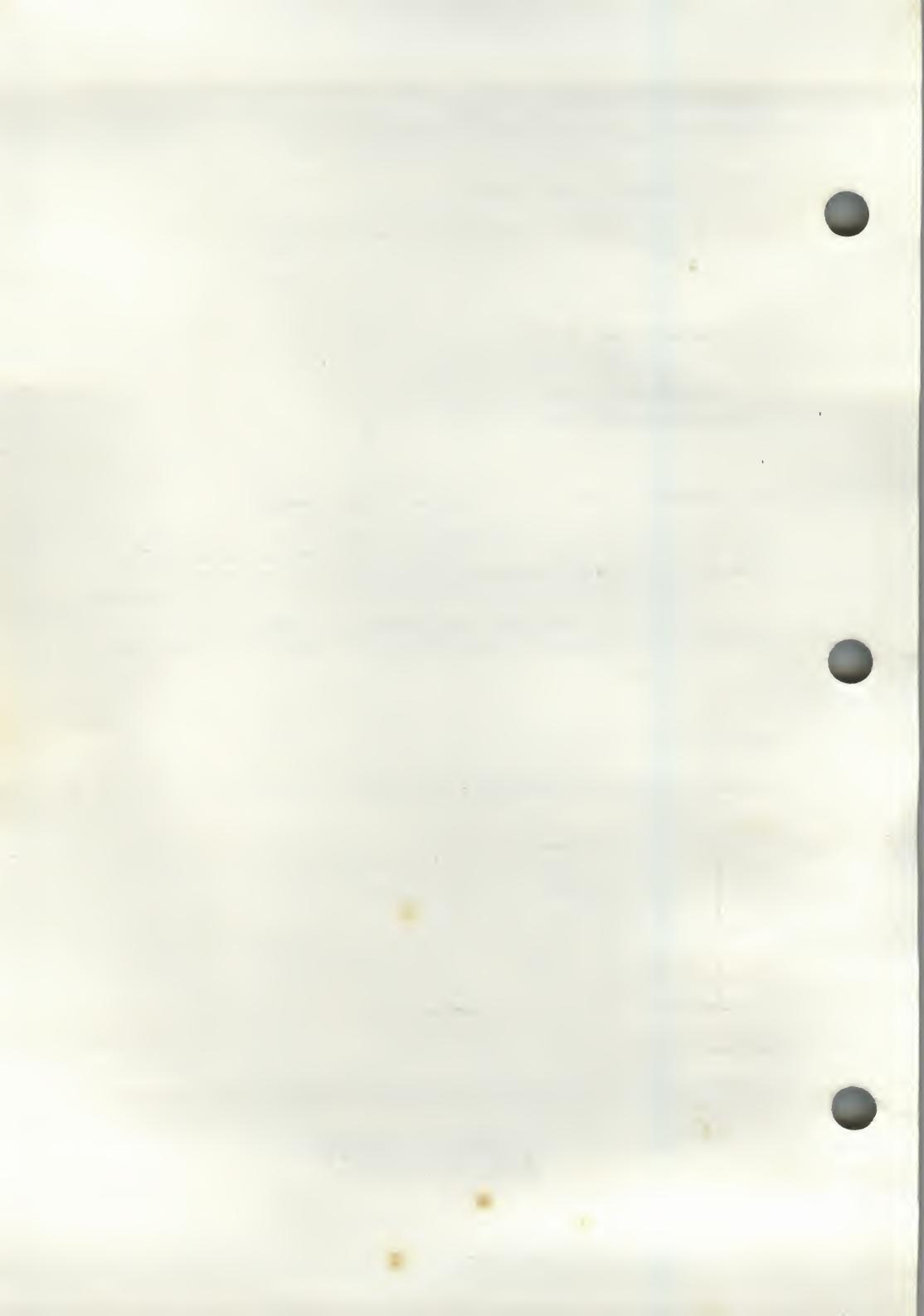
The last level completed on the attached form, Updating Status, indicates the pages to be added, removed or replaced, the number of pages included, and the Newsletter Code. Pages marked with an asterisk should be removed from the publication. The form should be filed at the back of the publication as a permanent record of amended pages.

Each amended page is identified by the Newsletter Code shown above.
Amended pages remain valid unless otherwise noted in a subsequent Newsletter.

Summary of Amendments:

1. Addition of description of GW-BASIC support for the Olivetti EGA Controller (OEC).
2. Amendment to OPEN COM statement.
3. Removal of LCOPY command.
4. Amendment of Random Access File description.
5. Amendment of LOCATE statement.

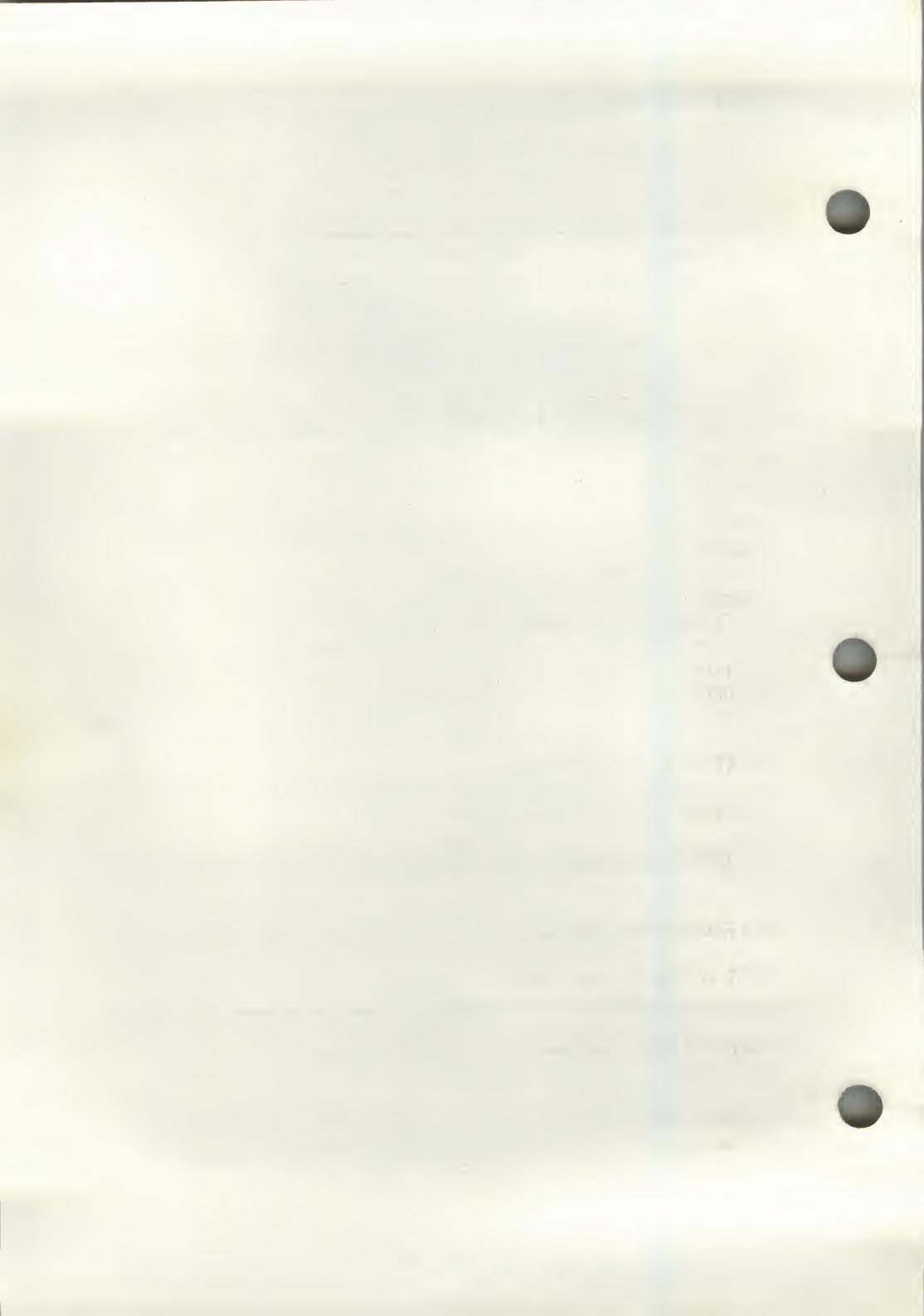
olivetti



UPDATING STATUS

LEVEL	DATE	UPDATED PAGES	PAGES	CODE
0				
1	25/6/87	Preface, Trademark, xiii, xviii, xix, about ch. 5, 5-1, about ch. 8, 8-3, 8-58, 8-61, 8-65, 8-137, 8-171, 8-188, 8-233, 8-300, about Ap. A, A-4, Appendix I.	33	4034491 D
2				
3				
4				
5				
6				
7				
8				
9				

Pages marked * must be suppressed



PREFACE

This manual is designed both as a user guide and as a reference manual for the MS GW-BASIC Interpreted language (Microsoft Ver. 3.2) available on the Olivetti Personal Computer.

SUMMARY

This manual is divided into two parts.

Part I, the first 7 chapters, cover: start-up, modes of operation, screen editing, general programming, disk I/O, graphics, machine language subroutines, event trapping, child processes and asynchronous communications.

Part II, Chapter 8, contains a detailed description of all commands, statements, and functions available, with examples for use.

Appendix A describes the differences between this version of GW-BASIC and versions prior to 3.2.

Appendix I describes the way in which the enhanced graphics capabilities of the Olivetti EGA Controller (OEC) can be utilized.

RELATED PUBLICATIONS

Installation and Operations Guide

MS-DOS User Guide

DISTRIBUTION: General (G)

FIRST UPDATE: July 1987

*Copyright © Microsoft Corporation
1980 ÷ 1986*

*Copyright © 1987 by Olivetti
All rights reserved*

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis - 10015 Ivrea — Italy

- OLIVETTI is a trademark of Ing. C. Olivetti & C., S.p.A.
- OLITALK is a trademark of Ing. C. Olivetti & C., S.p.A.
- OLITERM is a trademark of Ing. C. Olivetti & C., S.p.A.
- SORTP is a trademark of Ing. C. Olivetti & C., S.p.A.
- CLEO is a trademark of Phone 1 Inc.
- Dr. Logo is a trademark of Digital Research
- ETHERNET is a trademark of Xerox Corp.
- GEM COLLECTION is a trademark of Digital Research
- GEM DESKTOP is a trademark of Digital Research
- GEM DRAW is a trademark of Digital Research
- GEM GRAPH is a trademark of Digital Research
- GEM PAINT is a trademark of Digital Research
- GEM WORDCHART is a trademark of Digital Research
- GEM WRITE is a trademark of Digital Research
- GSS is a trademark of Graphic Software Systems, Inc.
- GSS-CHART is a trademark of Graphic Software Systems, Inc.
- GSS-DRIVERS is a trademark of Graphic Systems, Inc.
- GSS-KERNEL SYSTEM is a trademark of Graphic Software Systems, Inc.
- GSS-TOOLKIT is a trademark of Graphic Software Systems, Inc.
- GSX-86 is a trademark of Digital Research
- GW is a trademark of Microsoft Corp.
- IBM is a registered trademark of International Business Machines Corp.
- MICROSOFT is a registered trademark of Microsoft Corp.
- MS is a trademark of Microsoft Corp.
- OMNINET is a trademark of Corvus Systems Inc.
- p-System is a trademark of Softech Microsystem, Inc.
- PC-DOS is a trademark of International Business Machines Corp.
- PEACHPACK is a trademark of Peachtree Software International Ltd.
- TerminALL is a trademark of TOPIC System, Inc.
- TOPIC is a trademark of TOPIC System, Inc.
- UCSD and UCSD Pascal are registered trademarks of the Regent of the University of California
- UNIX is a trademark of Bell Laboratories
- Z80 is a registered trademark of Zilog Inc.
- Z8000 is a registered trademark of Zilog Inc.
- XENIX is a registered trademark of Microsoft Corp.

INPUT Statement	8-149
INPUT# Statement	8-152
INPUT\$ Function	8-153
INSTR Function	8-156
INT Function	8-158
IOCTL Statement	8-158
IOCTL\$ Function	8-161
KEY Statement	8-162
KEY(<i>n</i>) Statement	8-167
KILL Command	8-170
LEFT\$ Function	8-172
LEN Function	8-173
LET Statement	8-174
LINE Statement	8-175
LINE INPUT Statement	8-178
LINE INPUT# Statement	8-180
LIST Command	8-181
LLIST Command	8-183
LOAD Command	8-184
LOC Function	8-186
LOCATE Statement	8-187

LOCK Statement	8-189
LOF Function	8-191
LOG Function	8-192
LPOS Function	8-193
LPRINT and LPRINT USING Statement	8-194
LSET and RSET Statements	8-195
MERGE Command	8-197
MID\$ Function and Statement	8-198
MKDIR Command	8-201
MKI\$, MKS\$, MKD\$ Functions	8-203
NAME Command	8-204
NEW Command	8-207
OCT\$ Function	8-207
ON COM(<i>n</i>) GOSUB Statement	8-208
ON ERROR GOTO Statement	8-211
ON...GOSUB and ON...GOTO Statements	8-214
ON KEY(<i>n</i>) GOSUB Statement	8-215
ON PLAY(<i>n</i>) GOSUB Statement	8-219
ON TIMER (<i>n</i>) GOSUB Statement	8-222
OPEN Statement	8-224
OPEN COM Statement	8-232
OPTION BASE Statement	8-236

TAB Function	8-321
TAN Function	8-322
TIME\$ Statement and Function	8-323
TIMER Function	8-326
TIMER Statement	8-326
TRON/TROFF Commands	8-327
UNLOCK Statement	8-329
USR Function	8-331
VAL Function	8-332
VARPTR Function	8-333
VARPTR\$ Function	8-335
VIEW Statement	8-336
WAIT Statement	8-339
WHILE...WEND Statements	8-340
WIDTH Statement	8-342
WINDOW Statement	8-346
WRITE Statement	8-356
WRITE # Statement	8-357

A. DIFFERENCES BETWEEN THIS AND PREVIOUS VERSIONS

INTRODUCTION	A-1
CLS STATEMENT	A-1
COLOR STATEMENT	A-1

INPUT STATEMENT	A-1
INPUT # STATEMENT	A-1
LOCATE STATEMENT	A-2
SCREEN FUNCTION	A-2
SHELL STATEMENT	A-2
VIEW PRINT STATEMENT	A-2
WIDTH STATEMENT	A-3
SCREEN SCROLLING	A-3
/I SWITCH	A-3
FUNCTION KEY DISPLAY	A-3
SCREEN SCROLL	A-3
RANDOM ACCESS FILE DEFAULTS	A-4
LCOPY COMMAND	A-4

B. MATHEMATICAL FUNCTIONS

C. ASCII CHARACTER CODES

D. CONVERSION TABLES

E. CONVERSION OF PROGRAMS TO GW-BASIC

INTRODUCTION	E-1
STRING DIMENSIONING	E-1
LENGTH OF STRINGS	E-1

SUBSTRINGS	E-1
CONCATENATION	E-2
MAT FUNCTIONS	E-2
MULTIPLE ASSIGNMENTS	E-2
MULTIPLE STATEMENTS	E-3
PEEKs AND POKEs	E-3
IF...THEN...[ELSE...]	E-3
FILE I/O	E-3
GRAPHICS	E-4

F. MEMORY MANAGEMENT

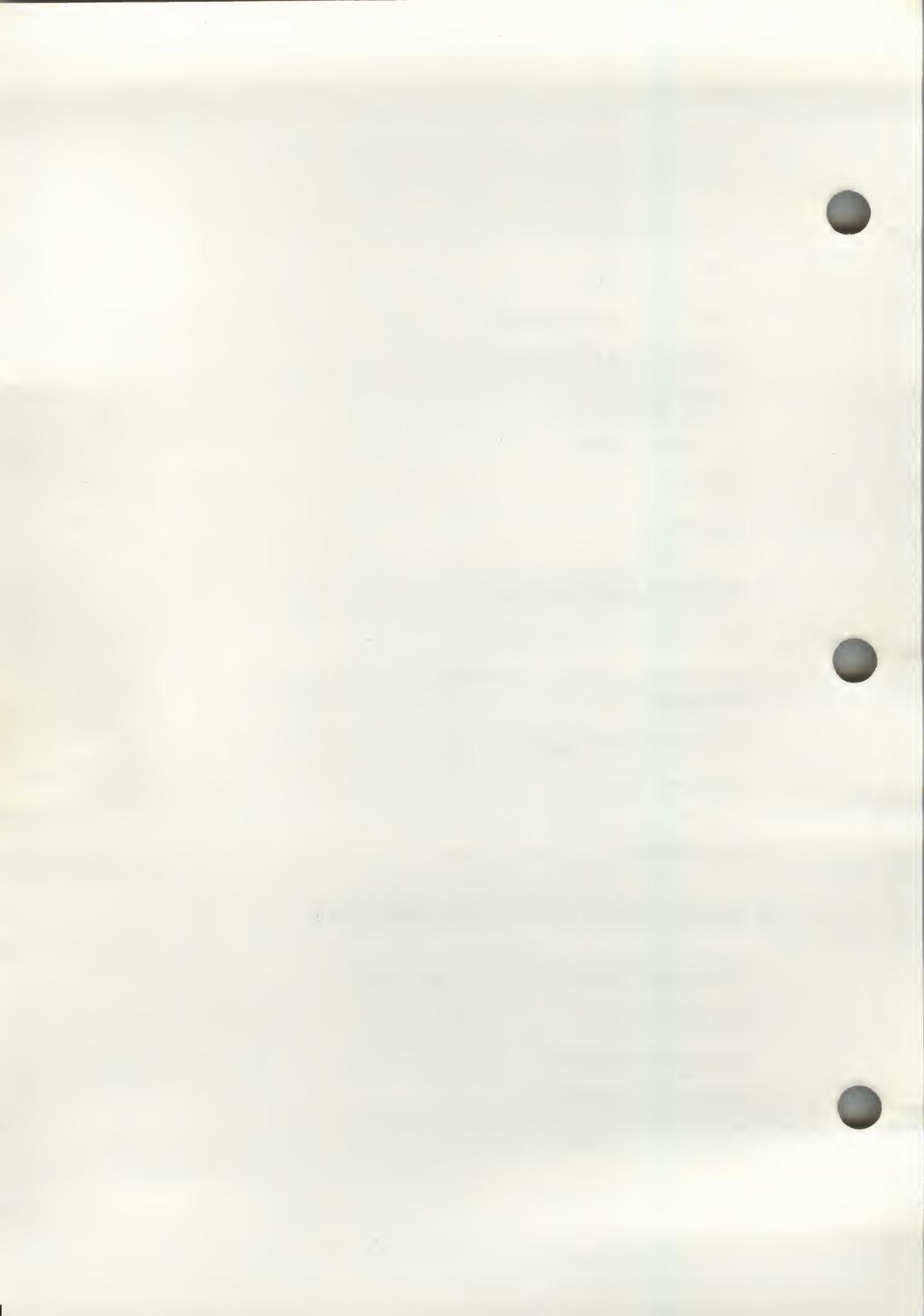
ALLOCATION OF VARIABLES	F-1
INTERNAL REPRESENTATION OF FLOATING POINT NUMBERS	F-3
FILE CONTROL BLOCK	F-4
MEMORY MAP	F-8

G. GLOSSARY

H. ERROR CODES AND ERROR MESSAGES

I. GW-BASIC SUPPORT FOR THE OEC

INTRODUCTION	I-1
COLOR STATEMENT	I-1
PALETTE STATEMENT	I-5
SCREEN STATEMENT	I-9



5. GRAPHICS

ABOUT THIS CHAPTER

GW-BASIC provides, under MS-DOS, a complete range of graphic features. This chapter introduces screen graphics including screen modes, color use, coordinates, the statements VIEW and WINDOW, and other graphics statements.

CONTENTS

SELECTING THE GRAPHICS ENVIRONMENT	5-1	COORDINATES	5-10
		ANIMATION TECHNIQUES	5-11
TEXT MODE	5-2		
GRAPHICS MODE	5-4		
MEDIUM RESOLUTION MODE	5-5		
HIGH RESOLUTION MODE	5-6		
SUPER RESOLUTION MODE	5-7		
WINDOWS AND VIEWPORTS	5-8		
TRANSFORMATION USING WINDOW AND VIEW	5-9		

SELECTING THE GRAPHICS ENVIRONMENT

The GW-BASIC language provides a set of graphics statements and functions which permit you to draw lines and points on the video using various colors. You must select the graphics environment in order to use these statements.

Upon initialization, the system is in Text Mode (SCREEN 0); you select the graphics environment using the SCREEN statement.

There are three different graphics modes you can select:

- Medium Resolution Mode (by entering SCREEN 1)
- High Resolution Mode (by entering SCREEN 2)
- Super Resolution Mode (by entering SCREEN 3)

(Note that additional enhanced graphics screen modes are available using the Olivetti EGA Controller (OEC), see Appendix I for details).

The screens differ in the number of points displayed and in the number of colors allowed.

The SCREEN statement also allows you to select the "active" and the "visual" page, in text mode (using the *apage* and *vpage* parameters). The active page is the page written to by subsequent output statements to the screen; the visual page is the one displayed on the screen.

The SCREEN statement must precede any I/O statements to the screen. The system assumes SCREEN 0,0,0,0 by default; this selects 80 columns Text Mode, and one display page.

You can also use more than one SCREEN statement to define different screen attributes for different sections of your program.

You can also change from one graphics mode to another by the WIDTH statement. The WIDTH statement allows you to set the screen width (in Text Mode) or change mode (in one of the graphics modes).

TEXT MODE (SCREEN 0)

In Text Mode you can display text, i.e. letters, numbers, and all special characters of the GW-BASIC character set. You can set the character foreground and background color using the COLOR statement. This statement also allows you to create blinking, reverse image, invisible, highlighted, and underscore characters.

Characters are displayed in horizontal lines from top (line 1) to bottom (line 25). Each line has 40 (or 80) columns. The WIDTH command allows you to select the number of columns.

The LOCATE statement positions the cursor on the screen. The cursor column and line coordinates are returned by the POS(0) and CSRLIN functions.

Characters are usually displayed, using the PRINT or PRINT USING statements, at the cursor position from left to right on each line, from line 1 to 24. When the cursor passes to line 25, lines 1 to 24 are moved one line up the screen.

Line 25 will usually display the Function Key values (see KEY statement in Chapter 8). To move the cursor to line 25 and display characters, use KEY OFF, then LOCATE and PRINT statements.

Multiple Display Page

Multiple display pages are allowed in Text Mode. Every statement that reads or writes from the screen is actually reading/writing from or to the active page. The visual page is the page that is shown on the screen, and may be different from the active page. This feature allows you to display a page, while writing another. The active and visual pages may be selected by the SCREEN statement.

COS Function	8-74	ERR and ERL Functions	8-111
CSNG Function	8-75	ERROR Statement	8-113
CSRLIN Function	8-76	EXP Function	8-115
CVI, CVS, CVD Functions	8-76	FIELD Statement	8-116
DATA Statement	8-78	FILES Command	8-119
DATE\$ Function and Statement	8-80	FIX Function	8-121
		FOR...NEXT Statements	8-122
DEF FN Statement	8-82	FRE Function	8-126
DEF SEG Statement	8-84	GET (COM files) Statement	8-127
DEF USR Statement	8-86	GET (Files) Statement	8-128
DEFINT / SNG / DBL / STR Statements	8-88	GET (Graphics) Statement	8-129
DELETE Command	8-89	GOSUB...RETURN Statements	8-132
DIM Statement	8-90	GOTO Statement	8-134
DRAW Statement	8-95	GWBasic Command	8-135
EDIT Command	8-101	HEX\$ Function	8-142
END Statement	8-102	IF...GOTO...ELSE and IF...THEN... ELSE Statements	8-143
ENVIRON Statement	8-103	INKEY\$ Function	8-146
ENVIRON\$ Function	8-105	INP Function	8-148
EOF Function	8-107	INPUT Statement	8-149
ERASE Statement	8-108		
ERDEV and ERDEV\$ Functions	8-109		

INPUT# Statement	8-152	LOCK Statement	8-189
INPUT\$ Function	8-153	LOF Function	8-191
INSTR Function	8-156	LOG Function	8-192
INT Function	8-158	LPOS Function	8-193
IOCTL Statement	8-158	LPRINT and LPRINT USING Statement	8-194
IOCTL\$ Function	8-161		
KEY Statement	8-162	LSET and RSET Statements	8-195
KEY(<i>n</i>) Statement	8-167	MERGE Command	8-197
KILL Command	8-170	MID\$ Function and Statement	8-198
LEFT\$ Function	8-172	MKDIR Command	8-201
LEN Function	8-173	MKI\$, MKSS\$, MKD\$ Functions	8-203
LET Statement	8-174		
LINE Statement	8-175	NAME Command	8-204
LINE INPUT Statement	8-178	NEW Command	8-207
LINE INPUT# Statement	8-180	OCT\$ Function	8-207
LIST Command	8-181	ON COM(<i>n</i>) GOSUB Statement	8-208
LLIST Command	8-183	ON ERROR GOTO Statement	8-211
LOAD Command	8-184		
LOC Function	8-186	ON...GOSUB and ON...GOTO Statements	8-214
LOCATE (Text) Statement	8-187	ON KEY(<i>n</i>) GOSUB Statement	8-215

COMMANDS, STATEMENTS AND FUNCTIONS

EDIT

Lets you change a specified program line.

FILES

Displays the names of files residing on the specified directory.

GWBASIC

Initializes GW-BASIC and the operating environment (GWBASIC is an MS-DOS command,.not a GW-BASIC command).

KILL

Deletes a disk file.

LIST

Lists the current program to the screen or a specified file or device.

LLIST

Lists the current program on the printer.

LOAD

Loads a program into memory from a specified drive and, optionally runs it.

MERGE

Merges the current program with a specified file previously saved in ASCII format.

MKDIR

Permits the creation of a new directory on a specified disk.

NAME

Changes the name of a disk file.

NEW

Deletes the current program and clears all variables, allowing you to enter a new program.

RENUM

Changes the line numbers of the current program.

RESET

Closes all open data files on all drives.

RMDIR

Removes a directory from a specified disk.

RUN

Runs the current program or loads a program from disk and runs it.

COMMANDS, STATEMENTS AND FUNCTIONS

Example

To read the data in a sequential file open for output or append, you must first CLOSE the file and then re-OPEN it in "I" mode.

```
100 CLOSE #1
110 OPEN "I", #1, "DATA"
```

CLS Statement



Erases all of the screen or a window.

CLS

Characteristics

CLS in text mode clears the entire screen to the current text background color. If a viewport has been defined, the current viewport only will be cleared to the graphics background color.

Remarks

CLS not only erases all or part of the screen, but also returns the cursor to the upper left-hand corner of the screen (in Text Mode).

If you are in Graphics Mode, CLS makes the "last referenced point" the center of the screen.

The screen can also be cleared by pressing **CTRL HOME** , or by modifying the screen mode using the SCREEN statement, or the width using the WIDTH statement.

Examples

```
5 COLOR 10,3
10 CLS
```

Clears the screen to cyan.

COLOR Statement (SCREEN 0)



Sets the text foreground and background colors in Text Mode.

See Appendix I for information on how this statement works with the Olivetti EGA Controller (OEC).

COLOR [foreground] [, background] [, dummy]]

Where

SYNTAX ELEMENT	MEANING
<i>foreground</i>	Is a numeric expression rounded to the nearest integer. It must be in the range 0 to 31. Values greater than 15 are interpreted modulo 16. It selects the character foreground color.
<i>background</i>	Is a numeric expression rounded to the nearest integer. It must be in the range 0 to 15, but it is interpreted modulo 8, thus only values from 0 to 7 are taken into consideration. It selects the character background color.

150 COLOR 15,1

This sets a high-intensity white on a blue background with a color screen, and a high-intensity white on a black background with a black and white screen.

Possible Errors

- If the COLOR statement ends in a comma (,), a "Missing operand" error is returned. For example:

COLOR 2,

is invalid.

- Any parameters outside the specified ranges will result in an "Illegal function call" error. In this case, previous values are retained.

COLOR Statement (SCREEN 1)

Defines the palette background and foreground colors in Medium Resolution Mode. In addition, the default graphics foreground and background colors, and the text foreground color can be defined. See Appendix I for information on how this statement works with the Olivetti EGA Controller (OEC).

COLOR [*background*][, *palette*][, *gforeground*][, *gbackground*][, *tforeground*]]]]

Where

SYNTAX ELEMENT	MEANING																
<i>background</i>	<p>Is a numeric expression rounded to the nearest integer. It must be in the range 0 to 31. Values greater than 15 are taken modulo 15. It represents the color code for the character background.</p> <table><tbody><tr><td>0 Black</td><td>8 Gray</td></tr><tr><td>1 Blue</td><td>9 Light Blue</td></tr><tr><td>2 Green</td><td>10 Light Green</td></tr><tr><td>3 Cyan</td><td>11 Light Cyan</td></tr><tr><td>4 Red</td><td>12 Light Red</td></tr><tr><td>5 Magenta</td><td>13 Light Magenta</td></tr><tr><td>6 Yellow</td><td>14 Light Yellow</td></tr><tr><td>7 White</td><td>15 High-intensity White</td></tr></tbody></table> <p>It also specifies the color for color number 0 that may be given with graphics statements. A value between 16 and 31 (interpreted modulo 15), specifies that foreground colors (text and color number 1, 2 and 3) are displayed with high intensity. It defaults to 0 (black).</p>	0 Black	8 Gray	1 Blue	9 Light Blue	2 Green	10 Light Green	3 Cyan	11 Light Cyan	4 Red	12 Light Red	5 Magenta	13 Light Magenta	6 Yellow	14 Light Yellow	7 White	15 High-intensity White
0 Black	8 Gray																
1 Blue	9 Light Blue																
2 Green	10 Light Green																
3 Cyan	11 Light Cyan																
4 Red	12 Light Red																
5 Magenta	13 Light Magenta																
6 Yellow	14 Light Yellow																
7 White	15 High-intensity White																
<i>palette</i>	<p>Is a numeric expression rounded to the nearest integer. It must be in the range 0 through 255. This selects one of two palettes. A graphics statement can specify a foreground color through a color number (0, 1, 2 or 3) which selects the desired color from the choice provided by the active palette. If the color number is 0, the color specified by <i>background</i> will be selected.</p> <table><tbody><tr><td>Palette</td><td>Color 0</td><td>Color 1</td><td>Color 2</td><td>Color 3</td></tr><tr><td>0</td><td><i>background</i> green</td><td>red</td><td>yellow</td><td></td></tr><tr><td>1</td><td><i>background</i> cyan</td><td>magenta</td><td>white</td><td></td></tr></tbody></table> <p>Palette 1, the default, is selected when <i>palette</i> is an odd number, otherwise palette 0 is selected.</p>	Palette	Color 0	Color 1	Color 2	Color 3	0	<i>background</i> green	red	yellow		1	<i>background</i> cyan	magenta	white		
Palette	Color 0	Color 1	Color 2	Color 3													
0	<i>background</i> green	red	yellow														
1	<i>background</i> cyan	magenta	white														

Examples

```
10 SCREEN 1,0  
20 COLOR 10,1,2,0
```

Sets the character background to light green, selects palette 1 (Cyan, Magenta, White), sets the graphics foreground to magenta, and graphics background to light green.

```
100 COLOR ,0
```

The character background stays light green and palette 0 (green, red, yellow) is selected.

COLOR Statement (SCREEN 2 and 3)

Defines the foreground color for text and graphics, in High and Super Resolution Modes. See Appendix I for information on how this statement works with the Olivetti EGA Controller (OEC).

COLOR [*foreground*]

Where

SYNTAX ELEMENT	MEANING																
<i>foreground</i>	<p>Is a numeric expression rounded to the nearest integer. It must be in the range 0 to 15. Larger values are interpreted modulo 16. This specifies the text and graphics foreground color according to the following:</p> <table><tbody><tr><td>0 Black</td><td>8 Gray</td></tr><tr><td>1 Blue</td><td>9 Light Blue</td></tr><tr><td>2 Green</td><td>10 Light Green</td></tr><tr><td>3 Cyan</td><td>11 Light Cyan</td></tr><tr><td>4 Red</td><td>12 Light Red</td></tr><tr><td>5 Magenta</td><td>13 Light Magenta</td></tr><tr><td>6 Yellow</td><td>14 Light Yellow</td></tr><tr><td>7 White</td><td>15 High-intensity White</td></tr></tbody></table> <p>The color upon entry into High Resolution or Super Resolution Mode is high-intensity white.</p>	0 Black	8 Gray	1 Blue	9 Light Blue	2 Green	10 Light Green	3 Cyan	11 Light Cyan	4 Red	12 Light Red	5 Magenta	13 Light Magenta	6 Yellow	14 Light Yellow	7 White	15 High-intensity White
0 Black	8 Gray																
1 Blue	9 Light Blue																
2 Green	10 Light Green																
3 Cyan	11 Light Cyan																
4 Red	12 Light Red																
5 Magenta	13 Light Magenta																
6 Yellow	14 Light Yellow																
7 White	15 High-intensity White																

Characteristics

Only one foreground color can be displayed at any one time. Changing the foreground color results in all current foreground being changed to the color specified. The background color is always black.

Example

```
10 SCREEN 2
20 COLOR 4
```

Results in High Resolution Mode with a foreground of red.

```
30 SCREEN 3
40 COLOR 5
```

Results in Super Resolution Mode with a foreground color of magenta.

COMMANDS, STATEMENTS AND FUNCTIONS

SYNTAX ELEMENT	MEANING
	<p>If this switch is present, then the maximum number of files is set to <i>number-of-files</i>. Each file requires 62 bytes for the File Control Block (FCB) plus 128 bytes for the data buffer. The data buffer size may be altered via the /S: switch. If the /F: option is omitted, the <i>number-of-files</i> is set to 3.</p>
	<p>The number of open files that MS-DOS supports depends upon the value of the FILES = <i>parameter</i> in the CONFIG.SYS file. It is recommended that FILES = 10 for GW-BASIC. Remember that the first 3 are taken by <i>stdin</i>, <i>stdout</i>, <i>stderr</i>, <i>stdaux</i>, and <i>stdprn</i>. One additional file handler is needed by GW-BASIC for LOAD, SAVE, CHAIN, NAME and MERGE. This leaves 6 for GW-BASIC File I/O, thus /F:6 is the maximum supported by MS-DOS when FILES = 10 appears in the CONFIG.SYS file.</p>
	<p>Attempting to OPEN a file after all the file handlers have been exhausted will result in a "Too many files" error.</p>
/S: <i>lrec</i>	<p>Is a switch that sets the maximum record length allowed with random files. If this switch is present, then the maximum record length is set to <i>lrec</i>. The record length option (<i>record-length</i>) on the OPEN statement cannot exceed this value. If the /S: option is omitted, the maximum record length defaults to 128 bytes. The maximum value permitted for <i>lrec</i> is 32767 bytes.</p>

SYNTAX ELEMENT	MEANING
/C: buffer-size	<p>If present, controls RS232 Communications. /C: disables RS232 support; any subsequent I/O attempts will result in a "Device Unavailable" error. Specifying /C:n allocates <i>n</i> bytes for the receive buffer for each RS232 port present. If the /C: option is omitted, GW-BASIC allocates 256 bytes for the receive buffer of each RS232 port present. 128 bytes are always allocated to the transmit buffer. The maximum value permitted for <i>buffer-size</i> is 32767.</p>
/M: [highest-memory] [,max-blocksize]	<p>When present, <i>highest-memory</i> sets the maximum number of bytes that will be used as GW-BASIC workspace. GW-BASIC will attempt to allocate 64K of memory for the data and stack segment. If machine language subroutines are to be used with GW-BASIC programs, use the /M: switch to set the highest memory location that GW-BASIC can use. When omitted or 0, GW-BASIC attempts to allocate all it can up to a maximum of 65536 bytes.</p> <p>In order to load programs above the GW-BASIC workspace you must use the optional parameter <i>max-blocksize</i> to reserve areas for the workspace and your programs. This is necessary if you intend to use the SHELL command. Failure to do so will result in COMMAND being loaded on top of your routines when a SHELL command is executed.</p> <p><i>max-blocksize</i> must be in Paragraphs (in multiples of 16). When omitted, &H1000 (4096) is assumed. This allocates 65536 bytes ($65536 = 4096 \times 16$) for GW-BASIC's Data and Stack segment. If you require 65536 bytes for GW-BASIC and 512 bytes for machine language subroutines, then use /M:,&H1010 (4096 paragraphs for GW-BASIC + 16 paragraphs for your routines).</p>

COMMANDS, STATEMENTS AND FUNCTIONS

Example

```
200 KILL "A:DATA1.DAT"  
300 KILL "C:DIR1\DIR2\PROG2.BAS"
```

Note that the filename must include the extension, if one exists. GW-BASIC does not supply the extension .BAS for the KILL command.

LEFT\$ Function

Returns a substring extracting a number of characters to the left of a given string, as specified by the *length* parameter.

LEFT\$(*string* , *length*)

Where

SYNTAX ELEMENT	MEANING
<i>string</i>	Is a string expression whose value is the string from which the substring is to be returned
<i>length</i>	Is an integer expression (from 0 to 255) which specifies the number of the characters to be returned.

Characteristics

If *length* is greater than LEN(*string*), the entire original string will be returned. If *length* = 0, the null string (length zero) will be returned.

Refer to the MID\$ and RIGHT\$ functions.

COMMANDS, STATEMENTS AND FUNCTIONS

For communications files LOC is used to determine if there are any characters in the input queue waiting to be read. The input queue can hold more than 255 characters (determined by the /C: switch). If there are more than 255 characters in the queue, LOC returns 255. Since strings are limited to 255 characters, this practical limit removes the need to test for string size before reading data into them. If fewer than 255 characters remain in the queue, the value returned by LOC depends on whether the device was opened in ASCII or binary mode.

In either mode, LOC will return the number of characters that can be read from the device. However, in ASCII mode, the low level routines stop queueing characters as soon as end-of-file is received. The character which indicates the end-of-file itself is not queued and cannot be read. An attempt to read the end-of-file will result in an "Input past end" error.

Example

```
100 IF LOC(2)> 100 THEN STOP
```

LOCATE Statement

Moves the cursor to the specified position on the active page. In Text mode LOCATE may also turn the cursor on and off and define the size of the cursor.

```
LOCATE [row][ , [column][ , [cursor][ , [start][ , stop]]]]
```

Where

SYNTAX ELEMENT	MEANING
<i>row</i>	Is the screen line number. A numeric expression returning an unsigned integer in the range 1 to 25.
<i>column</i>	Is the screen column number. A numeric expression returning an unsigned integer in the range 1 to 80, depending upon screen width.
<i>cursor</i>	Is a boolean value indicating whether the cursor is visible or not during program execution. A zero value turns the cursor off, a value of 1 turns the cursor on. This parameter can only be used in Text Mode.
<i>start</i>	Is a numeric expression whose integer value represents the cursor top (starting) scan line. Values allowed are between 0 and 7. This parameter can only be used in Text Mode.
<i>stop</i>	Is a numeric expression whose integer value represents the bottom (stop) cursor scan line. It has the same range of values as <i>start</i> , but must be greater than or equal to <i>start</i> ; if it is equal the cursor will be shown as a single scanline. If this parameter is omitted, and <i>start</i> is given, <i>stop</i> defaults to the value of <i>start</i> . This parameter can only be used in Text Mode.

Characteristics

If any of the parameters used with LOCATE are omitted, the current value is assumed.

COMMANDS, STATEMENTS AND FUNCTIONS

SYNTAX ELEMENT	MEANING
	O (odd) S (space)
<i>data</i>	Designates the number of data bits per byte. Valid entries are: 5, 6, 7 (default), or 8. If 8, the <i>parity</i> must be set to N.
<i>stop</i>	Designates the stop bit. Valid entries are: 1, 1.5, or 2. If omitted then 75 and 110 bps transmit two stop bits, all others transmit one stop bit.
RS	Suppresses RTS (Request To Send)
CS[t]	Controls CTS (Clear To Send)
DS[t]	Controls DSR (Date Set Ready)
CD[t]	Controls CD (Carrier Detect)
BIN	Opens the file in binary mode. BIN is selected by default, unless ASC is specified.
ASC	Opens the file in ASCII mode.
LF	Specifies that a linefeed is to be sent after a carriage return.

SYNTAX ELEMENT	MEANING
<i>mode</i>	<p>Is one of the following string expressions:</p> <p>OUTPUT Specifies sequential output mode</p> <p>INPUT Specifies sequential input mode</p> <p>If the <i>mode</i> expression is omitted, it is assumed to be random input/output. Random cannot, however, be explicitly chosen as <i>mode</i>.</p>
<i>filenum</i>	Is the number of the file to be opened.
<i>record-length</i>	Is the length of the records written to or read from a communications buffer. This value cannot be greater than the value fixed by the I/C: switch in the GWBASIC command. The default <i>record-length</i> for the receive buffer is 2 bytes. The length of the transmit buffer is 128 bytes.

Characteristics

The OPEN COM statement must be executed before an RS-232-C port can be used.

Any syntax errors in the OPEN COM statement will result in a "Bad file name" error message.

The *speed*, *parity*, *data*, and *stop* options must be listed in the order shown in the above syntax. The remaining options may be listed in any order, but they must be listed after the *speed*, *parity*, *data*, and *stop* options.

COMMANDS, STATEMENTS AND FUNCTIONS

SCREEN Function

Returns either the ASCII code (0-255) or the color number for the character at the specified screen location.

SCREEN(*row* , *column*[, *condition*])

Where

SYNTAX ELEMENT	MEANING
<i>row</i>	Is a numeric expression returning an unsigned integer in the range 1 to 25
<i>column</i>	Is a numeric expression returning an unsigned integer the range of which depends on the screen width
<i>condition</i>	Is a valid numeric, relational or logical expression returning a boolean result (0 or 1). If <i>condition</i> is given as non-zero, the color number for the character is returned instead of the ASCII code.

Characteristics

The SCREEN function returns zero if the system is in one of the graphics modes and the specified screen location contains graphics data.

Refer to Appendix C for a complete list of ASCII codes.

Examples

100 X = SCREEN (10,10) 'If the character at 10,10 is A then 65 is returned.

110 X = SCREEN (1,1,1) 'Returns the color number of the character in the upper left hand corner of the screen.

Errors

If you enter a value outside the above mentioned ranges, an "Illegal function call" error is returned.

SCREEN Statement

Allows you to pass from Text Mode to one of the graphics modes. It also allows you to enable/disable color and to select the active and visual page (in Text Mode). See Appendix I for an explanation of the other, enhanced graphics, screen modes that are available with the Olivetti EGA Controller (OEC).

SCREEN [mode] [, [burst] [, [apage] [, vpage]]]

**A. DIFFERENCES BETWEEN THIS AND
PREVIOUS VERSIONS**

ABOUT THIS APPENDIX

This appendix describes the difference between this version of GW-BASIC and previous versions.

CONTENTS

INTRODUCTION	A-1
CLS STATEMENT	A-1
COLOR STATEMENT	A-1
INPUT STATEMENT	A-1
INPUT # STATEMENT	A-1
LOCATE STATEMENT	A-2
SCREEN FUNCTION	A-2
SHELL STATEMENT	A-2
VIEW PRINT STATEMENT	A-2
WIDTH STATEMENT	A-3
SCREEN SCROLLING	A-3
/I SWITCH	A-3
FUNCTION KEY DISPLAY	A-3
SCREEN SCROLL	A-3
RANDOM ACCESS FILE DEFAULT	A-4
LCOPY COMMAND	A-4

WIDTH STATEMENT

In version 3.20 of GW-BASIC the *size* parameter may only be given the values 40 or 80.

SCREEN SCROLLING

For version 3.20 of GW-BASIC screen scrolling occurs when the current cursor is below the text window. For example:

```
10 KEY OFF
20 LOCATE 25,1 : PRINT "Sample"
30 GOTO 20
```

scrolls lines 1 through 24 for version 3.20, but no scrolling occurs in prior versions.

/I SWITCH

For version 3.20 of GW-BASIC, static File Control Blocks is the default condition (the /I switch is not supported). For versions of GW-BASIC prior to 3.20, static File Control Blocks were specified using the /I switch.

FUNCTION KEY DISPLAY

The function key display can only be removed using the KEY OFF statement with version 3.20.

SCREEN SCROLL

Version 3.20 of GW-BASIC will output, for non-disk I/O, an extra CR LF at the device width at the end of line output, and will mark the next field as the second field on the next line.

This program illustrates the current differences between version 3.20 and previous versions of GW-BASIC:

```
10 CLS
20 LOCATE 20,1 : PRINT "the line on which to print the X";
30 LOCATE 10,80 : PRINT "X"
```

When version 3.20 of GW-BASIC finds that the end-of-screen has been reached, it outputs an extra **CR LF** in addition to the one which follows the **PRINT "X"**. Thus, the output takes the form of the printed line, followed by a blank line, followed by "Ok". For previous versions of GW-BASIC, there is no intervening blank line.

RANDOM ACCESS FILE DEFAULTS

The default values that influence random access file I/O in version 3.20 programs, are the following:

- 1) The maximum number of files that may be open simultaneously is 3.
- 2) The maximum record length permitted is 128.

Values greater than these will produce error messages.

Both of the values may be altered by using the GWBASIC command's /F and /S switches. A full explanation of the use of the switches is given in the GWBASIC command description in Chapter 8.

LCOPY COMMAND

The **LCOPY** statement is not supported under version 3.20 of GW-BASIC.

I. GW-BASIC SUPPORT FOR THE OEC

ABOUT THIS APPENDIX

This appendix provides information for the user who has an Olivetti EGA Controller (OEC) installed and wishes to make use of its graphics capabilities via GW-BASIC.

CONTENTS

INTRODUCTION	I-1
COLOR STATEMENT	I-1
PALETTE STATEMENT	I-5
SCREEN STATEMENT	I-9

INTRODUCTION

There are three statements which make use of the enhanced graphics capabilities of the OEC. Two of the statements, COLOR and SCREEN, are extensions of existing statements, the other, PALETTE, is provided specifically for use with enhanced graphics. Each of the statements is described in this appendix with the syntax for use with the OEC functioning in enhanced graphics mode; otherwise the user should refer to Chapter 8 "Commands, Statements and Functions" for information on all graphics statements.

COLOR STATEMENT

Sets the display colors.

For Screen Mode 0:

COLOR [*foreground*][, [*background*][, *dummy*]]

For Screen Mode 1:

COLOR [*background*][, [*palette*]]

For Screen Modes 2 and 3:

COLOR [*foreground*]

For Screen Modes 7, 8, and 9:

COLOR [*foreground*][, [*background*]]

Characteristics

The use of the COLOR statement and its syntax for the various modes are explained below.

Screen Mode 0

Modifies the current default text foreground and background colors. The *foreground* color must be an integer expression in the range 0-31. It is used to determine the "foreground" color in text mode, which is the default color of text. Sixteen colors can be selected with the integers 0-15. A blinking version of each color can be selected by adding 16 to the color number, for example blinking color 7 is equal to 7 + 16 or 23. The *background* color must be an integer expression in the range 0-7, and is the color of the background for each text character. Blinking background colors are not permitted. If no parameters are specified, then the default color for foreground is white (7) and background is black (0).

The colors corresponding to the integer values are the following:

0 Black	8 Gray
1 Blue	9 Light Blue
2 Green	10 Light Green
3 Cyan	11 Light Cyan
4 Red	12 Light Red
5 Magenta	13 Light Magenta
6 Brown	14 Yellow
7 White	15 High-intensity White

Screen Mode 1

In this mode the COLOR statement has a unique syntax that includes a *palette* parameter that is an odd or even integer expression. This argument determines which set of display colors to use when displaying particular color numbers. The default color settings for the *palette* parameter are equivalent to the following:

COLOR, 0	'Same as the next three PALETTE statements
PALETTE 1,2	'Attribute 1 = color 2 (green)
PALETTE 2,4	'Attribute 2 = color 4 (red)
PALETTE 3,6	'Attribute 3 = color 6 (brown)

GW-BASIC SUPPORT FOR THE OEC

COLOR, 1	'Same as the next three PALETTE statements
PALETTE 1,3	'Attribute 1 = color 3 (cyan)
PALETTE 2,5	'Attribute 2 = color 5 (magenta)
PALETTE 3,7	'Attribute 3 = color 7 (white)

Note that a COLOR statement will override previous PALETTE statements.

Screen Mode 2

Assigns one of sixteen colors to the foreground color. The *foreground* must be an integer expression in the range 0-15, corresponding to the colors given above in the description of Screen Mode 0. Using the PALETTE statement, however, any of the 16 colors may be independently assigned to the background and foreground (attributes 0 and 1, respectively). Only two colors can be displayed at a time.

Screen 3

Assigns one of sixteen colors to the foreground color. The *foreground* must be an integer expression in the range 0-15, corresponding to the colors given above in the description of Screen Mode 0. The background is black. In this mode there is no enhanced graphics functionality and the PALETTE statement is not supported.

Screen Modes 7-9

In these modes the graphics background is given by the *background* color number, which must be in the valid range of color numbers appropriate to the screen mode. See the SCREEN and PALETTE statements for more details. The default line drawing color is specified by the *foreground* parameter.

The foreground color may be the same as the background color, making displayed characters invisible. The default background color is black, or color number 0, for all screen modes.

The PALETTE statement gives you flexibility in assigning different display colors to the actual color number ranges for the *foreground* and *background* colors. See the PALETTE statement later in this appendix for more details.

Possible Errors

Parameters outside valid numeric ranges result in "Illegal function call" error messages.

Examples

The following series of examples show COLOR statements and their effects in the various screen modes.

SCREEN 0
COLOR 1,2 'foreground 1, background 2

SCREEN 1
COLOR 1,0 'foreground 1, even palette number
COLOR 2,1 'foreground 2, odd palette number

SCREEN 2
COLOR 4 'foreground 4

SCREEN 3
COLOR 3 'foreground 3

SCREEN 7
COLOR 3,5 'foreground 3, background 5

SCREEN 8
COLOR 6,7 'foreground 6, background 7

SCREEN 9
COLOR 1,2 'foreground 1, background 2

PALETTE STATEMENT

Changes one or more of the colors in the palette.

PALETTE [*attribute*, *color*]

PALETTE USING *arrayname* (*arrayindex*)

Characteristics

A palette contains a set of colors with each color specified by an *attribute*. Each *attribute* is paired with an actual display *color*. This *color* determines the actual visual color on the screen, and is dependent on the setting of your screen mode.

PALETTE with no parameters sets the palette to a known initial setting. This setting is the same as the setting when colors are first initialized.

If parameters are given, *color* will be displayed whenever *attribute* is specified in any statement that uses a color. Any color changes on the screen occur immediately. Note that when graphics statements use *color* parameters, they are actually referring to attributes and not actual colors. PALETTE pairs attributes with actual colors.

For example, assume that the current palette consists of colors 0, 1, 2 and 3. The following DRAW statement:

```
DRAW''C3L100''
```

selects attribute 3, and draws a line of 100 pixels using the color associated with the attribute 3, in this case, also 3. If the statement

```
PALETTE 3,2
```

is executed, then the color associated with attribute 3 is changed to color 2. All text or graphics currently on the screen displayed using attribute 3 are instantaneously changed to color 2. All text or graphics subsequently displayed with attribute 3 will also be displayed in color 2. The new palette of colors will contain 0, 1, 2, 2.

With the USING option, all entries in the palette can be modified in one PALETTE statement. The *arrayname* parameter is the name of an integer array and the *arrayindex* specifies the index of the first array element in the *arrayname* to use in setting your palette. Each *attribute* in the palette is consecutively assigned to the respective *color* in the array. If the *color* parameter in an array entry is -1, then the mapping for the associated *attribute* is not changed. All other negative numbers are illegal values for *color*.

You can use the color parameter in the COLOR statement to set the default text color. (Remember that color arguments in other statements are actually what are called *attributes* in this discussion.) This color argument specifies the way that text characters appear on the display screen. Under a common initial palette setting, points colored with *attribute* 0 appear as black on the display screen. Using the PALETTE statement, you could, for example change the mapping of *attribute* 0 from black to white.

Remember, that a PALETTE statement executed without any parameters will assign all *attributes* to their default *colors*.

The following tables give *attribute* and *color* ranges and defaults for the various screen modes.

GW-BASIC SUPPORT FOR THE OEC

SCREEN MODE	ATTRIBUTE RANGE	COLOR RANGE
0	0-15	0-31
1	0-3	0-15
2	0-1	0-15
3	NA	0-15
7	0-15	0-15
8	0-15	0-15
9	0-15	0-63

Tab. I-1 Attribute and Color Ranges for Screen Modes

ATTRIBUTE VALUE FOR MODE			#	COLOR
1 2,3 0,7,8,9				
0	0	0	0	BLACK
		1	1	BLUE
		2	2	GREEN
		3	3	CYAN
		4	4	RED
		5	5	MAGENTA
		6	6	BROWN
		7	7	WHITE
		8	8	GRAY
		9	9	LIGHT BLUE
		10	10	LIGHT GREEN
1	11	11	11	LIGHT CYAN
	12	12	12	LIGHT RED
2	13	13	13	LIGHT MAGENTA
	14	14	14	YELLOW
3	1	15	15	HIGH-INTENSITY WHITE

Tab. I-2 Default Attributes and Colors for Screen Modes

Note that colors in the range 16-31 are blinking versions of colors 0-15.

Examples

The following changes all points colored with *attribute* 0 to *color* 2:

PALETTE 0,2

This does not modify the palette.

PALETTE 0, -1

The following changes each palette entry. All *attributes* are now mapped to display color zero (since the array is initialized to zero when it is first declared).

PALETTE A%(0)

The screen will appear as one single color. However, it will still be possible to execute any GW-BASIC statement.

The following example sets each palette entry to its appropriate initial display color:

PALETTE

SCREEN STATEMENT

Sets the Screen Mode.

SCREEN [*mode*][,[*burst*]][,[*apage*]][,[*vpage*]]

Characteristics

The *mode* parameter is an integer expression with legal values 0, 1, 2, 3, 7, 8 and 9. Other values are illegal.

Each of the Screen Modes is described in the following paragraphs.

SCREEN 0

- Text Mode only
- Either 40 x 25 or 80 x 25 text format with character box size of 8 x 14
- Assignment of 16 colors to any of 16 attributes
- 8 or 4 display pages

SCREEN 1

- 320 x 200 pixel medium resolution graphics
- 80 x 25 text format with character box size of 8 x 8
- Assignment of 16 colors to any of 4 attributes

SCREEN 2

- 640 x 200 pixel high resolution graphics
- 40 x 25 text format with character box size of 8 x 8
- Assignment of 16 colors to any of 2 attributes

SCREEN 3

- 640 x 400 pixel super resolution graphics
- 80 x 25 text format with character box size of 8 x 8
- Assignment of 16 colors to foreground

SCREEN 7

- 320 x 200 pixel medium resolution graphics
- 40 x 25 text format with character box size of 8 x 8
- 8 display pages
- Assignment of 16 colors to 16 attributes

SCREEN 8

- 640 x 200 pixel high resolution graphics
- 80 x 25 text format with character box size of 8 x 8
- 4 display pages

- Assignment of any of 16 colors to 16 attributes

SCREEN 9

- 640 x 350 pixel enhanced resolution graphics
- 80 x 25 text format with character box size of 8 x 14
- Assignment of any of 64 colors to 16 attributes
- 2 display pages

For composite monitors and TVs, the *burst* parameter is a numeric expression that is either true (non-zero) or false (zero). A value of zero disables color and permits display of black and white images only. A non-zero value permits color.

The *apage* and *vpage* parameters determine the active and visual memory pages. The active page is the area in memory where graphics statements are written, the visual page is the area of memory that is displayed on the screen. Animation can be achieved by alternating the display of graphics pages. The goal here is to display the already completed graphics output on the visual page, while executing graphics statements in one or more active pages. A page is displayed only when graphics output to that page is complete. Thus the following is typical:

SCREEN 7,, 1, 2 'work in page 1, show page 2

- Graphics output to page 1
- while viewing page 2

SCREEN 7,, 2, 1 'work in page 2, show page 1

- Graphics output to page 2
- while viewing page 1

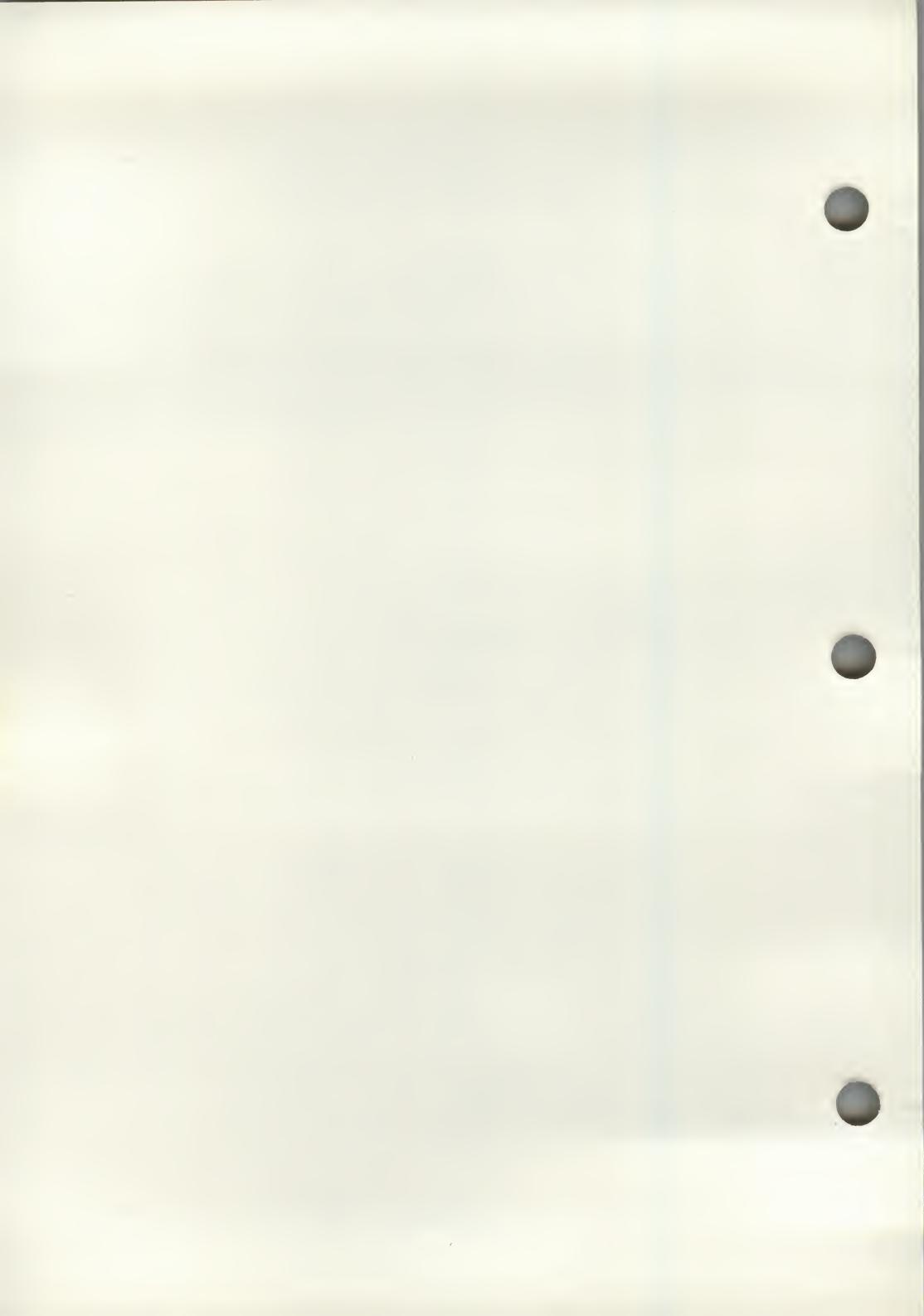
The number of pages available depends on the Screen Mode, as described in the following table.

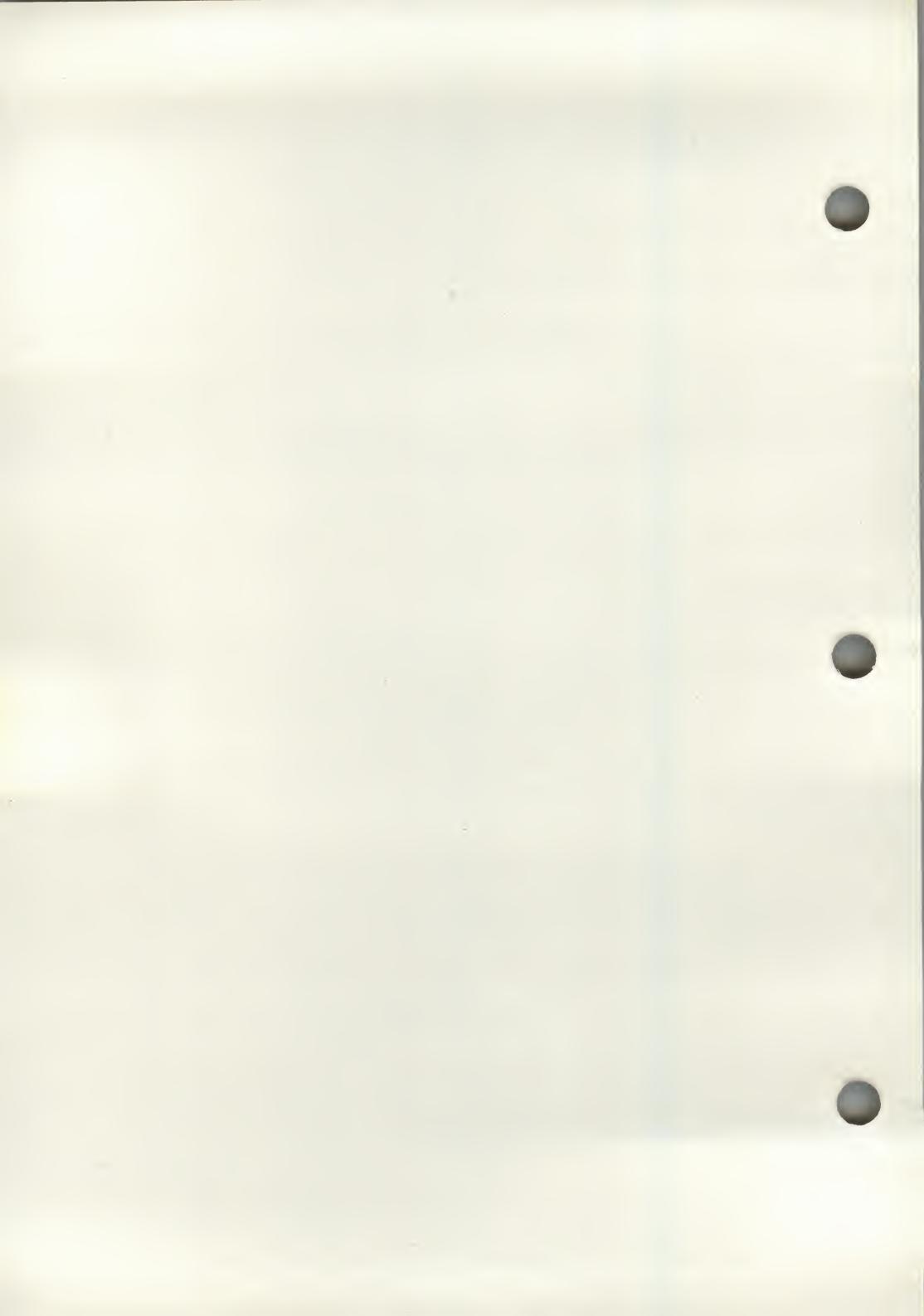
MODE	RESOLUTION	ATTRIBUTE RANGE	COLOR RANGE	PAGES	PAGE SIZE
0	40 column text 80 column text	0-15 0-15	0-31 0-31	8 4	2K 4K
1	320 x 200	0-3	0-15	1	16K
2	640 x 200	0-1	0-15	1	16K
3	640 x 400	NA	0-15	1	32K
7	320 x 200	0-15	0-15	8	32K
8	640 x 200	0-15	0-15	4	64K
9	640 x 350	0-15	0-63	2	128K

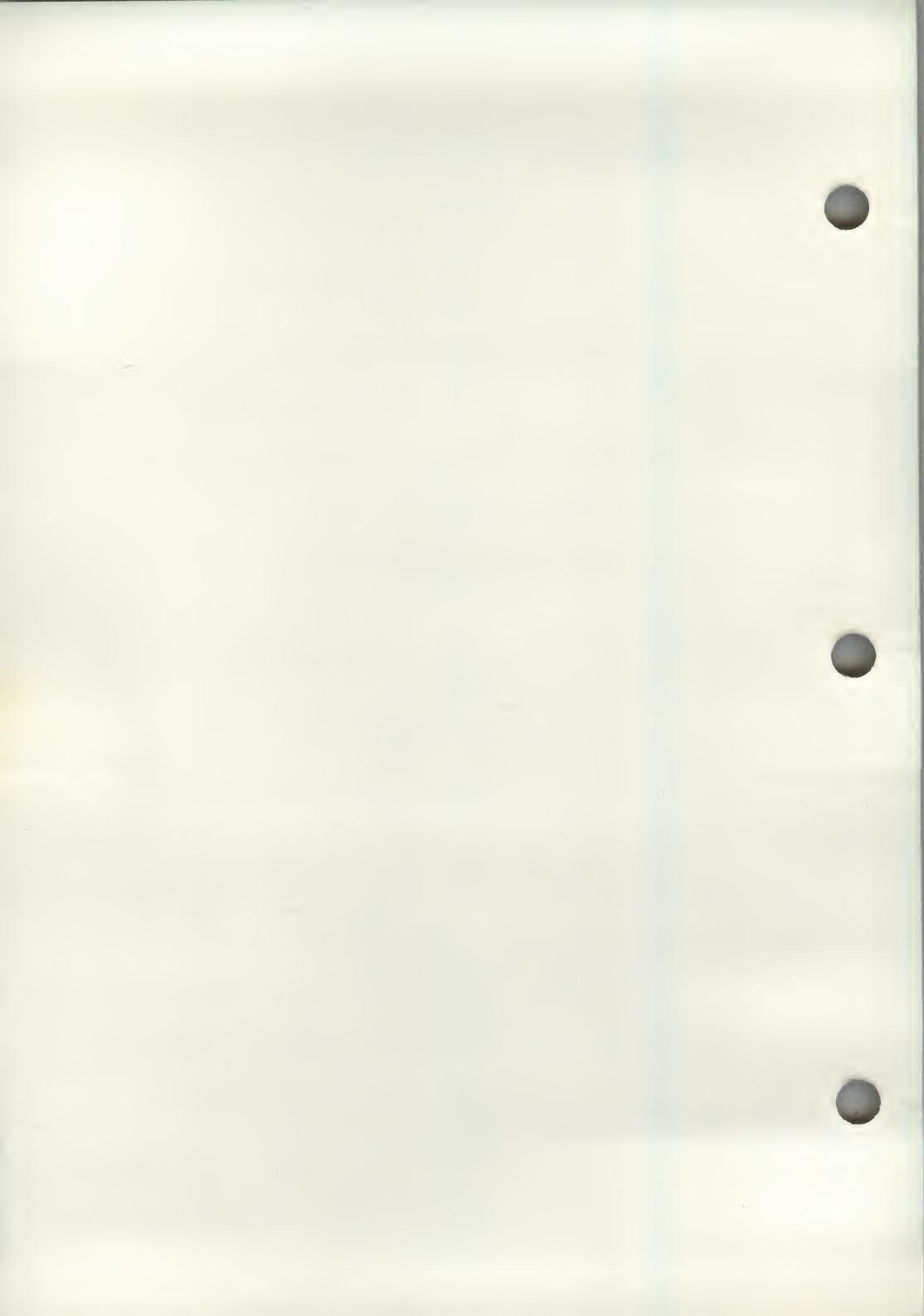
Tab. I-3 Screen Mode Specifications

Note that colors in the range 16-31 are blinking versions of the colors 0-15.

Different attribute and color settings exist. (See the PALETTE statement for a discussion of attribute and color number.)







FIAT

The information contained in this publication is intended to be of a general nature only.
The Fiat Company may at any time and from time to time, for technical or other
necessary reasons, modify any of the details or specifications of the product described
in this publication. To be sure of getting accurate, detailed and up-to-date information,
an intending buyer should consult his nearest Fiat Dealer or distributor or branch

Fiat Auto
Print. No. 603.06.275 - III/87 - 14. (1c) - Printed in Italy

a
Direzione Marketing e Commerciale - Assistenza Tecnica
10134 TORINO (Italy) - Corso E. Giambone, 33
8th. Ed. (1c) - Printed in Italy